

Milestone 1 Report - Team 03

Plans for testing and the process

- We use test driven development. Before we implemented a feature or fixed a bug, we first create a test for it.
- We use Junit 5.4.0 to tests.
- In order to get full test coverage, we used both MC/DC and Catalog-based-testing. The method used for different features depended on the feature itself. For example, if a function had many Boolean arguments, we used MC/DC to get the different possible combinations. We tried to use MC/DC as much as possible whenever the feature allowed for it.
- In general, we wanted to cover corner cases and the different possible input variations. We wanted to cover all types of outputs that could happen, and we also wanted to test the scenarios where an exception is thrown due to bad input. Therefore, we also try to test null input and missing input.
- We tried to test all functions, unfortunately it took a long time to get all right.
- Below are some examples of how we decided on what test cases to use:

Grep

MC/DC

Argument													
isCaseInsensitive		-	-	-	-	-	-	-	-	T	F	T	F
isCountOfLinesOnly		-	-	T	F	T	F	T	F	T	T	F	F
Input (l)	Not specified. Mock stdin.	Invalid file	-	-	-	I (contains aBcde)	I (contains aBcde)	I (does not contain aBcde)	I (does not contain aBcde)	"abcde"	"abcde"	"abcde"	"abcde"
Pattern		-	Null	""	""	"aBcde"	"aBcde"	"aBcde"	"aBcde"	"aBcde"	"aBcde"	"aBcde"	"aBcde"
Output		InvalidFileException	InvalidPatternException	0	""	1	Lines with "aBcde" joined with "\n"	0	""	0	1	""	"abcde"

Date

Type			Example		
Null			null		
Empty String			""		
String with values			"%m" "%m %d"		
Input	null	""	"%m-%d-%y"	"%H %M %S"	"year %y"
	Exception				

Wc

Argument										
IsBytes	-	T	T	F	F	T	T	F	T	F
IsLines	-	T	T	T	T	T	F	F	F	F
IsWords	-	T	T	T	F	F	F	F	T	T
Input (Inputstream)	Null	Empty input	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes	Some text (2 lines, 4 words and 12 bytes
Output	Exception	0 0 0	2 4 12	2 4	2	2 12	12	""	4 12	4

How to run the tests

- We use JDK 8.
- Our project has two tests folder that can be ran. The first is "test", which contains all tests for the features implemented by us in this milestone. The second one is "testEF2". This test folder contains tests for EF2, the features that we did not implement and only provided unit tests for.
- To run the tests, open the project in IntelliJ. It is then necessary to import the modules "org.junit.jupiter:junit-jupiter:5.4.0" and "org.mockito:mockito-all:1.10.19" using maven. Then right click the "test" folder and press "Run 'All Tests'". Alternatively, run the EF2 tests, right click the "testEF2" folder and press "Run 'All Tests'". This will run all tests in both folders. It is of course also possible to run the test classes individually.

Resources

- The resources used for testing can be found in the "resources" folder. This folder contains test directories and files that are used in various tests. For example, they are used in tests that test applications reading or outputting to a file.